Stat Comp Hw#2 Jason Rights

(informally) Due Thursday, 17 September, 1:00 PM

50 points total.

This assignment won't be submitted until we've covered Rmarkdown. Create R chunks for each question and insert your R code appropriately. Check your output by using the `Knit PDF` button in RStudio.

1. **Working with data** In the `datasets` folder on the course GitHub repo, you will find a file called `cancer.csv`, which is a dataset in comma-separated values (csv) format. This is a large cancer incidence dataset that summarizes the incidence of different cancers for various subgroups. (18 points)

    1. Load the data set into R and make it a data frame called `cancer.df`. (2 points)

```
cancer.df <- read.csv("https://raw.githubusercontent.com/fonnesbeck/Bios6301/master/datasets/cancer.csv
```

2. Determine the number of rows and columns in the data frame. (2)

```
dim(cancer.df)
```

```
## [1] 42120     8
```

42120 rows, 8 columns.

3. Extract the names of the columns in `cancer.df`. (2)

```
names(cancer.df)
```

```
## [1] "year"       "site"       "state"      "sex"        "race"
## [6] "mortality"  "incidence"  "population"
```

4. Report the value of the 3000th row in column 6. (2)

```
cancer.df[3000,6]
```

```
## [1] 350.69
```

5. Report the contents of the 172nd row. (2)

```
cancer.df[172,]
```

```
##     year                          site  state  sex  race mortality
## 172 1999 Brain and Other Nervous System nevada Male Black         0
##     incidence population
## 172         0      73172
```

6. Create a new column that is the incidence *rate* (per 100,000) for each row.(3)

```
cancer.df.new <- cbind(cancer.df,(cancer.df$incidence/cancer.df$population)*10000)
colnames(cancer.df.new) <- c(colnames(cancer.df),"incidencerate")
```

7. How many subgroups (rows) have a zero incidence rate? (2)

```
attach(cancer.df.new)
dim(cancer.df.new[which(incidencerate==0),])
```

```
## [1] 23191     9
```

There are 23191 subgroups with a zero rate.

8. Find the subgroup with the highest incidence rate.(3)

```
cancer.df.new[which(incidencerate==max(incidencerate)),]
```

```
##      year     site                state  sex  race mortality incidence
## 5797 1999 Prostate district of columbia Male Black     88.93       420
##      population incidencerate
## 5797     160821      26.11599
```

2. **Data types** (10 points)

    1. Create the following vector: `x <- c("5","12","7")`. Which of the following commands will produce an error message? For each command, Either explain why they should be errors, or explain the non-erroneous result. (4 points)

        ```
        max(x)
        sort(x)
        sum(x)
        ```

```
x <- c("5","12","7")
max(x)
```

```
## [1] "7"
```

```
sort(x)
```

```
## [1] "12" "5"  "7"
```

```
#sum(x)
```

Sum produces an error message because this is a character vector and it does not make sense to sum characters. Using "max" yeilds the entry that comes last when arranged in alphanumeric order, which in this case is 7 (as 7 comes after 5 or 1). Using sort puts them in alphanumeric order (12, 5, 7).

2. For the next two commands, either explain their results, or why they should produce errors. (3 points

        ```
        y <- c("5",7,12)
        y[2] + y[3]
        ```

```
y <- c("5",7,12)
#y[2] + y[3]
```

Y is a character vector because the character type is less flexible than the numeric type (thus, combining a character with numeric entries will make the whole vector character type). Since the "7" and "12" are characters, it does not make sense to add them, hence the error.

```
3. For the next two commands, either explain their results, or why they should produce errors. (3 points
```

```
        z <- data.frame(z1="5",z2=7,z3=12)
        z[1,2] + z[1,3]
```

```
z <- data.frame(z1="5",z2=7,z3=12)
z[1,2] + z[1,3]
```

```
## [1] 19
```

When you create z as a dataframe, it allows a combination of character and numeric entries, so the 7 and 12 are not converted to character like they are for vectors. Thus, it is fine to add them.

3. **Data structures** Give R expressions that return the following matrices and vectors (*i.e.* do not construct them manually). (3 points each, 12 total)

    1. $(1, 2, 3, 4, 5, 6, 7, 8, 7, 6, 5, 4, 3, 2, 1)$

```
c(seq(8),c(7:1))
```

```
##  [1] 1 2 3 4 5 6 7 8 7 6 5 4 3 2 1
```

```
2. $(1,2,2,3,3,3,4,4,4,4,5,5,5,5,5)$
```

```
x <- 1
for(i in 2:5){
  x <- c(x,replicate(i,i))
}
x
```

```
##  [1] 1 2 2 3 3 3 4 4 4 4 5 5 5 5 5
```

```
3. $\begin{pmatrix}
  0 & 1 & 1  \\
  1 & 0 & 1  \\
  1 & 1 & 0  \\
\end{pmatrix}$
```

```
x <- matrix(1,3,3)
diag(x) <- 0
x
```

```
##      [,1] [,2] [,3]
## [1,]    0    1    1
## [2,]    1    0    1
## [3,]    1    1    0
```

```
4. $\begin{pmatrix}
  1 & 2 & 3 & 4 \\
  1 & 4 & 9 & 16 \\
  1 & 8 & 27 & 64  \\
  1 & 16 & 81 & 256 \\
  1 & 32 & 243 & 1024  \\
\end{pmatrix}$
```

```
x <- matrix(NA,5,4)
for(i in seq(5)){
  for(j in seq(4)){
     x[i,j] <- j^(i)
  }
}
x
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1    2    3    4
## [2,]    1    4    9   16
## [3,]    1    8   27   64
## [4,]    1   16   81  256
## [5,]    1   32  243 1024
```

4. **Basic programming** (10 points)

    1. Let $h(x,n) = 1 + x + x^2 + \ldots + x^n = \sum_{i=0}^{n} x^i$. Write an R program to calculate $h(x,n)$ using a `for` loop. (5 points)

```
forloopforfour <- function(x,n){
  hxn <- 1
  for(i in seq(n)){
    hxn <- c(hxn,x^i)
  }
  print(sum(hxn))
}
```

```
1. If we list all the natural numbers below 10 that are multiples of 3 or 5, we get 3, 5, 6 and 9. The s
```

```
sum.natnumb <- function(mult1,mult2,N){
x <- seq(1:(N-1))
x.mult1 <- x[x%%mult1==0]
x.mult2 <- x[x%%mult2==0]
x <- c(x.mult1,x.mult2)
#print(x)
return(sum(as.numeric(unique(x))))
}
```

1. Find the sum of all the multiples of 3 or 5 below 1,000. (3, [euler1])

```
sum.natnumb(3,5,1000)
```

```
## [1] 233168
```

1. Find the sum of all the multiples of 4 or 7 below 1,000,000. (2)

```
sum.natnumb(4,7,1000000)
```

```
## [1] 178571071431
```

1. Each new term in the Fibonacci sequence is generated by adding the previous two terms. By starting w

```
x <- c(1,2)
  while(length(x[x%%2==0]) < 15){
    x <- c(x,sum(x[length(x)],x[length(x)-1]))
  }
print(sum(x[x%%2==0]))
```

```
## [1] 1485607536
```

Some problems taken or inspired by projecteuler.